Stereo Vision Based Velocity Control

Kenneth D. Owens, Jet Propulsion Lab
kowens@robotics.jpl.nasa.gov
818-354-8478

Abstract

The next phase of the unmanned ground vehicle development sponsored by OSD calls for intelligent velocity control to support cross-country driving speeds of 10 mph by day and 5 mph by night. To prevent vehicle damage at these speeds, rough terrain must be anticipated before inertial sensors on board have a chance to sense it. Using stereo range data collected in front of the vehicle along projected wheel tracks as input to a dynamic vehicle model, we develop a look ahead velocity control scheme which limits vehicle velocities to levels that keep the dynamic response of this model within given acceleration bounds.

1.0 Introduction

In order for high-speed autonomous cross-country vehicles to drive at terrain appropriate speeds, they must have the ability to understand the dynamic effects of the upcoming ground surface on their systems. Previous designs have relied upon either onboard inertial sensors or fixed suspension vehicle models. These approaches are effective at low speeds but at higher speeds are akin to driving blindfolded. What is needed to enable fast cross-country driving speeds is the human like ability to both sense the current vehicle condition and accurately interpret the vehicle response to upcoming terrain.

To enable this behavior, we focus on the development of a computationally fast vision based scheme for computing terrain appropriate velocity commands from a

dynamic suspension model. This model leads to a quadratic equation for estimating terrain appropriate velocity commands whose coefficients depend on the upcoming vertical road curvature. These coefficients are computed efficiently by taking advantage of the fact that the linear equation for the vertical road curvature, based on a cubic spline fit to the stereo range data, admits a Cholesky decomposition. This approach makes real time use of dynamic suspension models for velocity control possible.

This paper is divided into three main sections. In the first, we derive a quadratic equation for velocity control assuming that the road surface shape beneath the wheels is known. In the second, we discuss how the road surface shape is estimated from stereo range data using cubic spline approximations along anticipated wheel tracks. Finally, we show some field results concerning the velocity control of a roboticized HMMWV at the Jet Propulsion Laboratory.

2.0 Terrain Dependent Velocity Equation

To control vehicle velocity and limit the forces imposed on the vehicle while driving over rough terrain, a relationship between the vehicle's forward velocity and vertical accelerations is needed. One of the most basic ways to link these quantities is via the quarter vehicle model shown in figure 1.

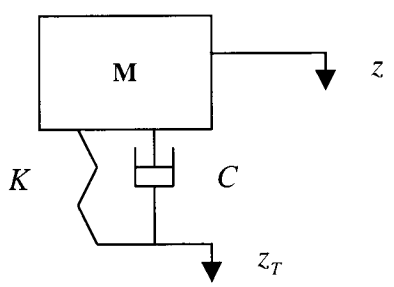


Figure 1

This model approximately describes the dynamic motion of a quarter vehicle suspension system where $M, K, C, z,$ and z_T are the quarter vehicle mass, the effective suspension spring constant, the effective suspension damping, and the displacements from equilibrium of the mass M and tire axle. The differential equation describing the motion of this system is well know

$$M\frac{d^2z}{dt^2} = -K(z - z_T) - C\frac{d}{dt}(z - z_T) \tag{2.1}$$

It is clear from this equation that if the relative displacement and its derivative are known, the vertical acceleration of the vehicle can be computed. Thus the following derivation will focus on solving for the relative displacement and its derivative.

If we subtract $M\frac{d^2z_T}{dt^2}$ from both sides of equation (2.1) we can write

$$\frac{d^2r}{dt^2} + \frac{C}{M}\frac{dr}{dt} + \frac{K}{M}r = \frac{d^2h}{dt^2}$$

where the relative displacement, r, and the road height, h, are given by $r = z - z_T, h = -z_T$. We can solve this equation for the relative displacement and its derivative, by rewriting it as a first order equation using $x_1 = r, x_2 = dr/dt$,

$$\frac{dx_1}{dt} - x_2 = 0 \quad \text{and} \quad \frac{dx_2}{dt} + \frac{K}{M}x_1 + \frac{C}{M}x_2 = \frac{d^2h}{dt^2}$$

Or in matrix notation, $\dot{x} + Ax = \begin{pmatrix} 0 \\ \frac{d^2h}{dt^2} \end{pmatrix}$ where $A = \begin{pmatrix} 0 & -1 \\ \frac{K}{M} & \frac{C}{M} \end{pmatrix}, x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, which has the solution $x(t) = e^{-At}x_0 + \int_0^t e^{A(\tau - t)}\begin{pmatrix} 0 \\ \frac{d^2h}{dt^2} \end{pmatrix}d\tau$. If we concern ourselves only with the effects of surface forcing on the vehicle ignoring initial conditions, we have

$x(t) = \int_0^t e^{A(\tau - t)} \begin{pmatrix} 0 \\ \frac{d^2 h}{dt^2} \end{pmatrix} d\tau$. This solution assumes that we know the road forcing as a function of time. However, in the field, the road forcing is computed not as a function of time but of distance. Changing variables from time, t, to distance, s, and assuming a constant vehicle velocity, U, we can rewrite the solution to our system as $x(l) = U \int_0^l e^{\frac{A}{U}(s-l)} \begin{pmatrix} 0 \\ \frac{d^2 H}{ds^2} \end{pmatrix} ds$ where $h(t) = H(s(t))$ and we used the chain rule to deduce that the second derivative with respect to time can be written as $\frac{d^2 h(t)}{dt^2} = U^2 \frac{d^2 H}{ds}$.

The solution of this equation is found by making several approximations. First, since the range data discrete, we will only look for solutions where there is data. i.e. $x_n = U \int_0^{l_n} e^{\frac{A}{U}(s-l_n)} \begin{pmatrix} 0 \\ \frac{d^2 H}{ds^2} \end{pmatrix} ds$. To approximate this equation, we assume that the time, $\Delta l_n / U$, to travel between range data points is small, allowing us to approximate the matrix exponential and write

$$x_n = (I - \frac{\Delta l_n A}{U}) x_{n-1} + U \int_{l_{n-1}}^{l_n} (I - \frac{\Delta l_n A}{U}) \begin{pmatrix} 0 \\ \frac{d^2 H}{ds^2} \end{pmatrix} ds$$

which is a difference equation of the form $x_n = \alpha_n x_{n-1} + \beta_n$ where

$$\alpha_n = (I - \frac{\Delta l_n A}{U}),\ \beta_n = U \begin{pmatrix} 0 \\ F_1^n \end{pmatrix} + A \begin{pmatrix} 0 \\ F_2^n \end{pmatrix},\ F_1^n = \int_{l_{n-1}}^{l_n} \frac{d^2 H}{dt^2} ds, \text{ and } F_2^n = \int_{l_{n-1}}^{l_n} (s - l_n) \frac{d^2 H}{dt^2} ds.$$

The exact solution of this difference equation can be found by inspection and is $x_n = \beta_n + \sum_{k=1}^{n-1} \prod_{j=k+1}^{n} \alpha_j \beta_k$ which leads quickly to highly nonlinear equations in the vehicle velocity U. However, is we consider only the two most significant terms, we get the simplified approximate solution $x_n = \alpha_n \beta_{n-1} + \beta_n$. Substituting for α, β and A, the

solution may finally be written as $x_n = \begin{pmatrix} -f_2^n + \frac{\Delta l_n}{U}\frac{C}{M} f_3^n \\ Uf_1^n + \frac{C}{M} f_2^n + \frac{\Delta l_n}{U}(\frac{K}{M} - (\frac{C}{M})^2) f_3^n \end{pmatrix}$

where $f_1^n = F_1^n + F_1^{n-1}$, $f_2^n = F_2^n + F_2^{n-1} + \Delta l_n F_1^{n-1}$, and $f_3^n = F_2^{n-1}$.

Recall that equation (2.1) implies that the vehicle vertical acceleration is $a = -\frac{K}{M} r - \frac{C}{M}\frac{dr}{dt}$ where r and dr/dt are the first and second components of x_n. Substituting these components into equation (2.1) and collecting terms implies that the vehicle vertical acceleration is given by $a = c_2 f_2^n - Uc_1 f_1^n - \frac{\Delta l_n}{U} c_3 f_3^n$ where $c_1 = \frac{C}{M}$, $c_2 = \frac{K}{M} - (\frac{C}{M})^2$, and $c_3 = \frac{K}{M}\frac{C}{M} - (\frac{C}{M})^3$.

To develop a driving strategy so that the vertical acceleration remains within the limits

$$| a | = | c_2 f_2^n - Uc_1 f_1^n - \frac{\Delta l_n}{U} c_3 f_3^n | \le a_{MAX} \tag{2.2}$$

we choose U so that

$$| c_2 f_2^n | + | Uc_1 f_1^n | + | \frac{\Delta l_n}{U} c_3 f_3^n | \le a_{MAX} .$$

This equation would seem to be impossible to satisfy for small U but, when one recalls that $\Delta l_n / U = \Delta t_n$, this apparent difficulty goes away. Likewise, the vehicle vertical acceleration will remain within bounds if we choose U such that

$$| c_2 | f_2^{\max} + Uc_1 f_1^{\max} + \frac{\Delta l_{\max}}{U} c_3 f_3^{\max} \le a_{MAX} \tag{2.3}$$

where $f_i^{\max} = \max_n | f_i^n |, \Delta l_{\max} = \max_n | \Delta l_n |$. Equation (2.3) implies

$$U^2 c_1 f_1^{\max} + U(| c_2 | f_2^{\max} - a_{\max}) + \Delta l_{\max} c_3 f_3^{\max} \le 0$$

and, since the coefficient of U^2 is positive, all of the velocities between the roots of

$$U^2 c_1 f_1^{\max} + U(| c_2 | f_2^{\max} - a_{\max}) + \Delta l_{\max} c_3 f_3^{\max} = 0 \tag{2.4}$$

satisfy inequality (2.3). The largest possible speed, which we use as a velcoity command, would therefore be the maximum of the two roots of equation (2.4).

Road Surface Estimation

As we saw in the last section, the critical information needed to estimate how upcoming terrain will effect vehicle acceleration is the second derivative of the road height. As we will see in this section, this information is an intermediate byproduct of cubic spline estimation of the road surface making it unnecessary to actually compute the functional shape of the road itself. In this section we will discuss how the second derivatives of the road surface along the anticipated wheel tracks of the vehicle are estimated.

The anticipated wheel tracks are assumed to be parallel to the two dimensional path generated by the onboard path planner, with stereo range data projected onto them making them three dimensional. The road roughness component is extracted from these paths by removing the low frequency component of the terrain. Once the road surface roughness along the wheel paths has been estimated, this data is transformed into road surface roughness as a function of distance traveled which is approximated by cubic splines. The cubic spline derivation is standard and follows [4] but is summarized here to show its relationship to the proposed efficient numerical solution of the resulting spline equation.

Since we are using cubic splines, their second derivative must be linear and for distances $l\varepsilon[l_{j-1}, l_j]$ can be written as

$$S''(l) = M_{j-1}(l_j - l)/\Delta l_j + M_j(l - l_{j-1})/\Delta l_j$$

where M_j are the values of the spline second derivative at the knots l_j. The entire second

derivative of the spline function for $l\varepsilon[l_0, l_N]$ can be written as

$$S''(l) = g^T(l)M \quad \text{where} \quad M^T = (M_1, M_2, \cdots, M_N) \quad \text{and} \quad g^T(l) = (g_1(l), g_2(l), \cdots, g_N(l))$$

where $g_i(l)$ are the functions necessary to form the spline and have the form

$$\begin{aligned} g_j(l) = & \ (l - l_{j-1})/\Delta l_j \quad \text{for} \quad l\varepsilon[l_{j-1} - l_j] \\ & -(l - l_{j+1})/\Delta l_{j+1} \quad \text{for} \quad l\varepsilon[l_j - l_{j+1}] \end{aligned}$$

Integrating $S''(l)$ twice, and requiring that $S(l)$ be continuous at the knots, leads to the equation

$$S(l) = \frac{1}{6}M_{j-1}[(l_j - l)^3/\Delta l_j] + \frac{1}{6}M_j[(l - l_{j-1})^3/\Delta l_j] + \frac{1}{6}[6y_{j-1} - M_{j-1}(\Delta l_j)^2][(l_j - l)/\Delta l_j] + \frac{1}{6}[6y_j - M_j(\Delta l_j)^2][(l - l_j)/\Delta l_j]$$

Further, by requiring that $S'(x)$ is continuous at the knots, we get the relationship

$$\frac{1}{6}\Delta l_j M_{j-1} + \frac{1}{3}(\Delta l_j + \Delta l_{j+1})M_j + \frac{1}{6}\Delta l_{j+1}M_{j+1} = \frac{1}{\Delta l_{j+1}}(y_{j+1} - y_j) - \frac{1}{\Delta l_j}(y_j - y_{j-1})$$

or in matrix notation

$$BM = Dy \tag{3.1}$$

where

$$B = \begin{bmatrix} \frac{1}{3}(\Delta l_1 + \Delta l_2) & \frac{1}{6}\Delta l_2 & \cdots & 0 \\ \frac{1}{6}\Delta l_2 & \frac{1}{3}(\Delta l_2 + \Delta l_3) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{1}{6}\Delta l_{N-1} \\ 0 & \cdots & \frac{1}{6}\Delta l_{N-1} & \frac{1}{3}(\Delta l_{N-1} + \Delta l_N) \end{bmatrix},$$

$$D = \begin{bmatrix} \frac{1}{\Delta l_1} & -(\frac{1}{\Delta l_1} + \frac{1}{\Delta l_2}) & \frac{1}{\Delta l_2} & 0 & \cdots & 0 \\ 0 & \frac{1}{\Delta l_2} & -(\frac{1}{\Delta l_2} + \frac{1}{\Delta l_3}) & \frac{1}{\Delta l_3} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \frac{1}{\Delta l_{N-1}} & 0 \\ 0 & \cdots & 0 & \frac{1}{\Delta l_{N-1}} & -(\frac{1}{\Delta l_{N-1}} + \frac{1}{\Delta l_N}) & \frac{1}{\Delta l_N} \end{bmatrix}$$

and $y^T = [y_0 \quad y_1 \quad \cdots \quad y_N]$ is the vector of spline ordinates.

The above relationship (3.1) between the spline ordinates, spline second derivatives and knot spacing holds for all cubic splines. However, we would like to use a smoothing spline which allow us to weight each point relative to its importance. On way to formulate the solution of this problem is to look for splines satisfying (3.1) that minimize data misfit and variations in spline curvature. Mathematically, this can be done by minimizing the squared sum of the spline function and data misfit. i.e.

$$K = \int M^T g(l) g(l)^T M dl + (y - f)^T \Lambda (y - f)$$

where $\Lambda = diag[\lambda_1 \quad \lambda_2 \quad \cdots \quad \lambda_N]$ is a diagonal weighting matrix and

$f = [f_1, \quad f_2, \quad \cdots \quad , f_N]$ is the vector of data ordinates. This equation is rewritten using (3.1) to get

$$K = y^T D^T (B^{-1})^T \int g(l) g(l)^T dl \; B^{-1} D y + (y - f)^T \Lambda (y - f)$$

Differentiating with respect to y, using the fact that $\int g(l) g(l)^T dl = B$ [4] and equating to zero gives

$$D^T (B^{-1})^T D y + \Lambda (y - f) = 0$$

Substituting the relationship (3.1) into this equation, simplifies it finally to the equation that must be solved for the smoothing spline second derivatives

$$(D \Lambda^{-1} D^T + B) M = D f \qquad (3.2)$$

Standard texts on numerical analysis explain that any positive definite symmetric matrix admits a Cholesky decomposition without row exchanges [5]. i.e. if P is a positive definite symmetric matrix, P can be written as $P = LL^T$ where L is a lower triangular matrix. This allows the system $Px = b$ to solved by solving the backwards and forwards

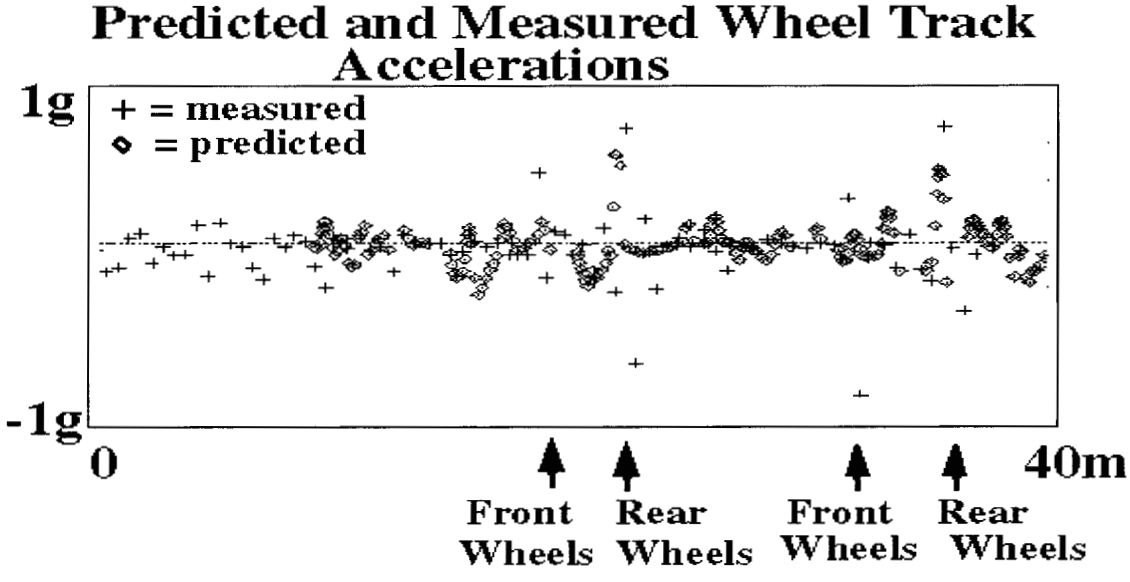
substitutions, $Lz = b$ *and* $L^T x = z$. To show that $D\Lambda^{-1}D^T + B$ has a Cholesky decomposition, we note that $D\Lambda^{-1}D^T + B$ is symmetric by observation and to demonstrate that it is positive definite, we must show that $x^T(D\Lambda^{-1}D^T + B)x > 0$ for all $x \neq 0$. Since B is symmetric, diagonally dominant and has positive diagonal elements, it is positive definite [6]. i.e. $x^T Bx > 0$ for all $x \neq 0$. $x^T(D\Lambda^{-1}D^T)x$ is at least positive since $x^T(D\Lambda^{-1}D^T)x = (D^T x)^T \Lambda^{-1}(D^T x) \geq 0$ for all $x \neq 0$ by inspection. Thus, $D\Lambda^{-1}D^T + B$ is positive definite, symmetric and therefore admits a Cholesky decomposition without row exchanges. Further, since $D\Lambda^{-1}D^T + B$ is banded, with two nonzero off diagonals above and below the main diagonal, L is also banded further reducing computation.

Thus, we are able to solve (3.2) for the second derivatives on the spline fit without solving for the spline itself, which additionally reduces computation. These second derivatives are used directly to compute the coefficients of the quadratic equation for velocity control discussed in the previous section.

4.0 Application

At Aberdeen Proving Grounds (APG), wooden 4X4 boards and rocks were run over at two and five miles per hour by a HMMWV while gathering range images and vehicle state. The large rock to the far right and in the far distance in the image below were not part of this test and were avoided.

Vertical accelerations several meters in front of the vehicle were predicted from the quarter vehicle dynamics model and compared with inertial accelerations measured when the vehicle traversed the same terrain. The following plot shows the results of hitting first the boards and then the rocks.



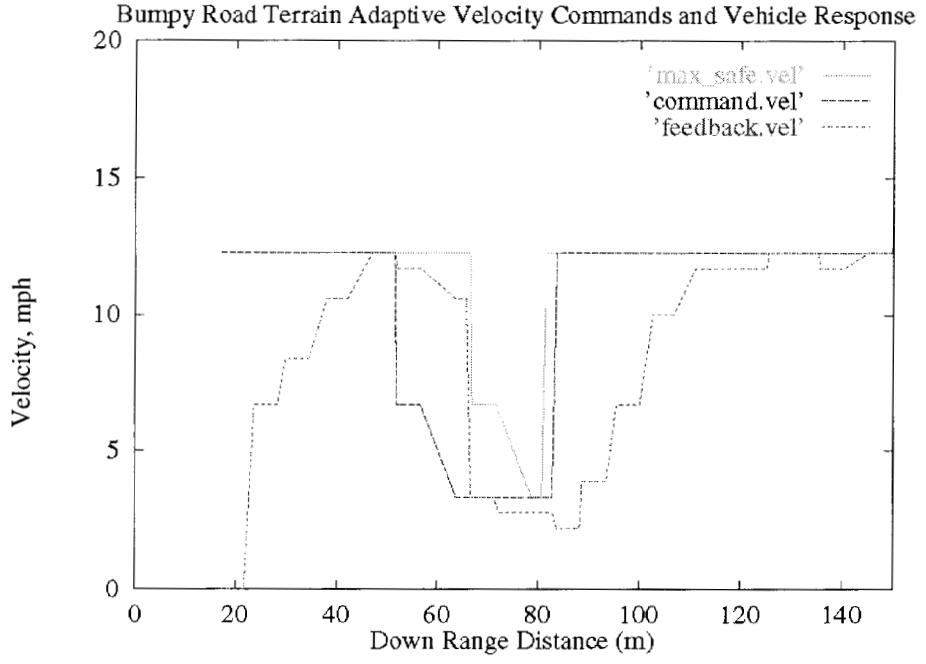
As seen in the above graph, the first spike in the inertial acceleration when the front wheels of the vehicle hit the 4X4's is not predicted. This is due to the fact that the quarter vehicle model is situated at the rear wheels. However, the next positive spike when the

rear wheels hit the 4X4's is well modeled in both location and amplitude. The downward spike is not well modeled since the camera cannot see behind the 4X4's. For velocity control purposes, the accelerations on the far side of the bumps can be assumed to be of the approximately same magnitude as those accelerations produced on the leading edge of the bump. Thus, accelerations from the leading edges of bumps provide a good indicator of terrain roughness. Finally, the HMMWV ran over the rocks similar model results showing an un-modeled first spike when the front wheels hit the rocks followed by a well modeled acceleration spike when the rear wheels hit the rocks.

The driving strategy must slow the vehicle for rough terrain and keep the velocity low until the rough area is cleared. The strategy used to accomplish this behavior was to save a short history of velocity estimates corresponding to the terrain from the vehicle's current position to a sufficient reaction distance and then to command a velocity that is the minimum of these estimates. This has the effect of slowing the vehicle for rough terrain and preventing the vehicle from speeding up until it is safe.

To test this strategy in the field, we drove the JPL HMMWV over cardboard bumps on a dirt road during the day and night collecting range data with CCD cameras by day and Amber Radiance I infrared cameras by night. These bumps were approximately 12 inches high and spaced at about 1 meter intervals for 10 meters. We set the maximum driving speed to 12 mph and allowed the vehicle to control its velocity. During these tests, the velocity commands were derived by repeatedly solving equation (2.2) with higher and higher velocities until the preset acceleration limit was exceeded. This was done because the technique for direct velocity estimation developed in equation (2.4) had not yet been developed.

In the following plot we show, the maximum safe driving speed estimated by the vehicle, the vehicle response and the commanded velocity. We see that the vehicle performed exactly as desired when it determined that running over the bumps at maximum speed was unsafe and that it was necessary to slow down in order to proceed. This control strategy also caused the vehicle to keep going slow as it traversed the bumps, in spite of seeing clear road ahead. When the vehicle was finally clear of the bumps, it sped up as desired.



Conclusion

A technique was demonstrated for controlling the driving speed of an autonomous vehicle based on look ahead range and a vehicle dynamics model. This method extracted the road height from range images and used it to force a quarter vehicle model. An equation was derived for the maximum safe driving speed that keeps the model

accelerations within limits. Finally, a velocity control strategy was devised which commands a driving speed that is the minimum over a short velocity history of the maximum possible driving speeds. This technique slowed the JPL test vehicle for rough terrain and sped up only after the vehicle was on smooth ground. In future work, texture or color could be added to determine if objects such as bushes or clumps of grass can be driven over. Also, the dynamic model could be extended to include pitch, roll and tire response if more fidelity is required.

Acknowledgments

This work was performed at the Jet Propulsion Laboratory, California Institute of Technology, sponsored by the Joint Robotics Program of the Office of the Secretary of Defense and managed by the Army Research Lab. We would also like to thank Dr. Larry Matthies for insisting during several conversations that direct velocity estimation was possible.

References

[1] W. Flugge, Handbook of Engineering Mechanics, McGraw-Hill, 1962.

[2] T.G. Gillespie, Fundamentals of Vehicle Dynamics, SAE, 1992.

[3] R.A. Jones, Validation Study of Two Rigid Body Dynamics Computer Models, Department of the Army, Waterways Experiment Station, Report GL-92-17, 1992.

[4] P. Lancaster, K. Salkauskas, Curve and Surface Fitting, Academic Press, 1988.

[5] H. R. Schwarz, Numerical Analysis, Wiley 1989.

[6] H. R. Schwarz, H. Rutishauser, E. Stiefel, Numerik symmetrischer Matrizen. 2 Aufl. Teubner, Stuttgart, 1972.